REMARKS

Claims 1-48 were pending in the present application. Claims 2 and 18 were cancelled. Claims 1, 4-14, 16, 17, 20-30, 32-34, 36-46, and 48. Accordingly, claims 1, 3-17, and 19-48 are now pending in the application.

Claims 1-48 stand rejected under 35 U.S.C. §103(a) as being anticipated by Liencres et al. (U.S. Patent No. 5,434,993) (hereinafter " Liencres") in view of Chandrasekaran et al. (U.S. Patent No. 6,970,872) (hereinafter "Chandrasekaran"). Although Applicant respectfully traverses at least portions of this rejection, Applicant has amended the claims for clarification and to expedite allowance.

Applicant's claim 1 recites a system comprising in pertinent part

> "a node including an active device, an interface to an inter-node network, a system memory, and an address network and a data network that is separate from the address network, coupling the active device, the interface, and the system memory;
>
> …
>
> wherein in response to receiving from the active device an address packet initiating a transaction to gain an access right to a coherency unit, the system memory is configured to send data corresponding to the coherency unit to the active device dependent on memory response information associated with the coherency unit, wherein the memory response information includes information used to derive global access state information for the coherency unit;
>
> wherein if the transaction cannot be satisfied within the node, the system memory is configured to forward a report corresponding to the address packet to the interface via the data network using a data packet, wherein in response to the report, the interface is configured to send the additional node a coherency message requesting the access right via the inter-node network." (Emphasis added)

The Examiner asserts on page 2 in the present Office action "the memory and the active device are part of element 32 in Liencres, which is connected to element 33."

However as shown in FIG. 3a, Liencres clearly shows that the memory eluded to by the Examiner is a cache memory 37, which is not coupled to the bus 33 but to the cache controller 35. In addition, in Applicant's specification and drawings it is clear the claimed memory is a system memory and not a cache memory. Furthermore, even though Liencres clearly shows a system (main) memory coupled by a memory bus 25 to each node, the Examiner has attempted to say the cache memory 37 is analogous to Applicant's claimed memory.

However, although Applicant completely disagrees with this analogy, Applicant has amended the claims clarify. Thus, Applicant submits element 33 cannot be the address network and the data network as recited in Applicant's claim 1 because it only couples the bus controller 31 to the processor cache controller 35. It does not couple the active device, the interface, and the **system** memory as recited in claim 1. Further, Liencres does not teach or disclose an address network and a data network that is separate from the address network, since element 33 is a simple cache bus (as disclosed by Liencres below).

Liencres also discloses at col. 7

> "Read Transactions
> When a memory request by the processor 21 cannot by fulfilled by the data in the processor cache memory 37, the processor cache controller 35 sends a read request packet across the cache bus 33 to the bus cache controller 31. The bus cache controller 31 proceeds to broadcast a corresponding read request packet across the memory bus 25. The read transaction initiated by the bus cache controller 31 consists of two packets: a read request packet sent by the bus cache controller 31 on the memory bus 25 and a read reply packet sent by another device on the memory bus. The read request packet contains the address of the memory requested by the processor cache controller 35 and is broadcast to all entities on the memory bus 25. A device on the memory bus 25 that contains the requested memory address responds to the read request packet with a read reply packet containing the subblock which includes the requested memory address. The read reply packet is generally issued by the main memory 23 except when the desired memory address is "owned" by another processor subsystem 20. In that case, the processor subsystem that owns the subblock must generate a read reply packet with the requested data." (Emphasis added)

In addition from the foregoing, Liencres is teaching the cache controller sending a read request that contains the address to the bus controller, which broadcasts the read request to all devices on the memory bus. Applicant fails to see how this teaches the system memory sending the data dependent upon memory response information in contrast to the Examiner's assertions. Accordingly, Applicant submits Liencres does not teach or disclose "wherein in response to receiving from the active device an address packet initiating a transaction to gain an access right to a coherency unit, the system memory is configured to send data corresponding to the coherency unit to the active device dependent on memory response information associated with the coherency unit, wherein the memory response information includes information used to derive global access state information for the coherency unit" particularly since Liencres teaches the cache control sends the cache directory information and not the system memory.

The Examiner acknowledges Liencres does not teach "wherein if the transaction cannot be satisfied within the node, the system memory is configured to forward a report corresponding to the address packet to the interface, wherein in response to the report, the interface is configured to send the additional node a coherency message requesting the access right via the inter-node network."

However, the Examiner asserts Chandrasekaran teaches the limitation at col. 6 lines 25-36. The Examiner asserts by ***paraphrasing*** Chandrasekaran "When another node writes out data, it sends out a report stating the latest write time for that data. The read data is invalid once its timestamp comes before the latest write time. The node, now having an invalid read data, will have to request the updated data from the additional node. It would have been obvious… to employ optimistic reading of data using write time validity checking so that reads could be employed when another node has exclusive access but hasn't yet written the data." Applicant respectfully disagrees with the Examiner's application of the Chandrasekaran art to Applicant's claims.

More particularly, Chandrasekaran is directed to optimistic reads and write time validity checking. Chandrasekaran discloses at col. 6, lines 25-36

> "In an embodiment using the first type of validity checking, the time that the optimistic read is started is compared to the latest time that the data block was written by any of the other nodes. If the read was started after the last write, the read is valid. This can be determined even before the read is finished, but involves the writing node publishing its write time to the other nodes. A node can publish its write time in any way, such as by broadcasting the write time to the other nodes, by storing the write time and responding to requests from other nodes, or by sending the write time to a lock manager. This type of validity checking is called "write-time" validity checking herein."

From the foregoing description, Applicant fails to see how this applies to Applicant's claim 1. Specifically, it appears to Applicant from the above disclosure and from the Examiner's own paraphrasng Chandrasekaran is teaching a given node broadcasting or somehow publishing its write time for a write of data. Then some other node does an optimistic read of the data, and if that read occurs earlier in time than the write, the read data would be invalid. The write time may be kept by a lock manager which returns the write validity information. Chandrasekaran teaches at col. 8 lines 56-66

> "If it is determined in step 250 that the particular block returned in step 240 is not valid, then control passes to step 260. In step 260, the operation to retrieve the particular block from disk or a remote cache is started again, based on permission received in step 230. Because permission has been received in step 230 before performing step 260, the data block received in response to step 260 will be valid. In embodiments in which permission is not received in step 230, such as embodiments in which a message denying permission to access the particular data block is received in step 230, step 260 is delayed until permission is received."

From the foregoing Applicant submits Chandrasekaran does not teach the limitations recited in Claim 1. To the contrary, Chandarsekaran is merely teaching a lock manager sending the write time and not the system memory, and retrying the read when permission is received. This is not the same as sending a report to the interface which forwards that report to another node.

From the foregoing, Applicant submits neither Liencres nor Chandrasekaran taken either singly or in combination teaches or suggest the combination of features recited in Applicant's claim 1. Accordingly, Applicant submits claim 1, along with its dependent claims patentably distinguishes over Liencres and over Liencres in view of Chandrasekaran for the reasons given above.

Applicant's claims 17 and 33 recite features that are similar to the features recited in claim 1. Thus Applicant submits claims 17 and 33, along with their respective dependent claims, patentably distinguish over Liencres for at least the reasons given above.

CONCLUSION

Applicant submits the application is in condition for allowance, and an early notice to that effect is requested.

If any fees are due, the Commissioner is authorized to charge said fees to Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C. Deposit Account No. 501505/5181-99901/SJC.

Respectfully submitted,

/ Stephen J. Curran /

Stephen J. Curran
Reg. No. 50,664
AGENT FOR APPLICANT(S)

Meyertons, Hood, Kivlin, Kowert, & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: September 7, 2007